STRAWMAN PROPOSAL FOR AN APPROACH TO THE PROBLEM OF MISSING INPUT DATA SETS FOR MODIS PROCESSES

by

Richard P. Cember
MODIS Science Data Support Team
7501 Forbes Blvd., Suite 103
Seabrook, MD 20702

rcember@ltpmail.gsfc.nasa.gov
301-352-2148

October 7, 1996

One of the major unresolved issues for MODIS as we go from Version 1 into Version 2 is how we plan to handle the problem of missing inputs to MODIS processes, either MODIS products which are not available when needed as inputs, or ancillary data sets which are not available when needed.

Some MODIS products and ancillary data sets can be considered as possible alternates for each other when the desired data set is not available. However, both the use and the non-use of alternates raise a number of issues which go beyond the domain of ancillary data *per se*. The purpose of this informal note is to highlight what I think some of the questions are by putting up a "strawman" proposal for a MODIS approach to this issue.

Operational Problem: Lack of a specified input

I see three basic responses to a runtime lack of a specified input:

Response 1:

Stop processing until the input is available again. This would occur for cases where the input is critical to the output and there is no acceptable substitute.

Response 2:

Do without the input and keep processing. This would occur where the input is desirable but not critical and there is either (a) no acceptable substitute, or (b) available substitutes do not confer sufficient advantage to be worth bothering with.

Response 3:

Use a substitute input and keep processing. This is for cases in which substitutes are acceptable, available, and confer enough advantage over doing without so as to be worth bothering with.

In Response 3, the input is either a MODIS product or an ancillary data stream, and the substitute could be either a MODIS product, an ancillary data stream, or a static ancillary data set (such as a climatology).

The proposal is to make it a MODIS goal to avoid Response 1 whenever possible, but to leave the choice of when in fact to invoke Response 1 up to the individual SCF team producing the product. Similarly, when Response 1 is not chosen, the choice between Response 2 and Response 3 would be up to the SCF. The only formal requirement would be that for each product, each team formally specify under what conditions of missing inputs it would be necessary to stop its processing. This formal specification would allow SDST to develop a tree of "go-no go" dependencies for MODIS processing.

A question that would occur under the proposed scenario is, "When do we decide that a late input product is to be considered as not available? How long do we wait if we don't know what the status of an input product is?"

Role of SDST

If alternate ancillary data sources are to be turned to, they have to be readable by the executing process' code, like any other ancillary data source. Indeed, under the proposed scenario, neither the project as a whole nor SDST needs, operationally speaking, to be concerned with the distinction between primary and alternate data sources. All that is needed under this scenario is the normal ancillary data activity, in which the particular SCF team works with SDST to get the necessary files identified and added to the ancillary data requirements given to ECS.

"Consumer disclosure" for end users of products

Changes in the inputs to a product are something that users of a product might well want to know about. How will they be told in a convenient manner? Fred Patt of SDST has suggested the solution for this, based on Seawifs practice. Inputs can be listed in the metadata for each product file. For example, if a particular product used MODIS ozone to make one output file and TOMS ozone to make another, that would be reflected in the metadata for each of those files. Sufficiently curious end users could then trace the chain of inputs as far back as they wish to do so, but end-user concerns in regard to consistency of inputs would not dictate when products are produced. Products would be made whenever scientifically possible.

A question that would have to be resolved if this approach of *caveat emptor* metadata were implemented is, "How far down toward the root of the processing tree should a product's inputs be identified?" One level down would be sufficient to enable a motivated user unambiguously to trace the input history all the way back to the radiances, but would a more informative approach be better? Or would it lead to an unmanageable clutter of input tracing? Would it be desirable to have a yes-no flag in the metadata which indicates whether all the normal inputs were present?

Code-writing overhead

For a particular input, a hierarchy of substitute inputs creates the necessity of writing more reading code. This is an overhead cost for science, about which each science team would be free to make its own judgment. Scientists in general want to make good products; they also have limited resources. Each science team would be free to decide how best to use its resources to get the best overall scientific result.

It is noted above that alternate ancillary data sets would be just like other ancillary data sets. It is likely that when alternates are specified, the end result will be that for many ancillary data sets there will be more than one user. Science teams could cut down on their code-writing overhead by sharing routines. For example, if a team that uses NMC GDAS input chooses DAO as a backup source, they might get a DAO reading routine from a team that has already chosen DAO as a primary source, and modify it as needed to fit their particular needs.